

How To Use CLI Agent

이수안, 김재성

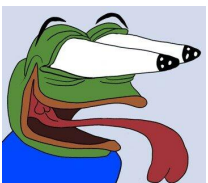
Why CLI? 왜 CLI를 써야 할까?

1. 그야... 실험이 오래 걸리니...



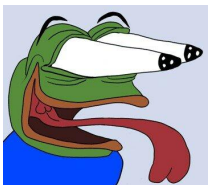
다들 오류 해결 못해서 **10시간**씩
오류 붙잡고 있던 일을 방지 가능

2. 반복 실험 자동화 가능



CLI Agent를 통해 결과 나올 때까지 기다릴 필요 없이
자기가 알아서 기다렸다 결과 보고 **다음 실험** 알아서 진행

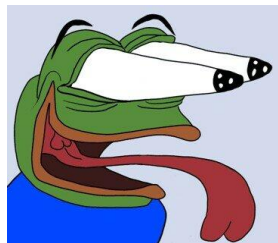
3. 모든 역할을 동시에



각각의 SubAgent를 통해 모든 일을 “**딸깍**”으로 동시에 처리 가능
논문쓰면서, 실험하면서, 가설 세우면서... 등등

Why CLI? 왜 CLI를 써야 할까?

```
➤ Shell export CUDA_VISIBLE_DEVICES=1 && /home/dilab05/anaconda3/envs/cognitive_momentum/bin/python final_experiment_transformers.py [in /home/dilab05/work_directory/김재성/cognitive_mome...
⚙ Implementing and Integrating (esc to cancel, 8h 18m 4s) <- 8시간동안 쓰즈로 일처리중
> | Type your message or @path/to/file
~/work_directory/김재성 no sandbox (see /docs) gemini-2.5-pro (58% context left) | ✖2 errors (ctrl+o for details)
```



우리가 해야 할 실험을 스스로 하고 결과 분석,
시각화, 재실험을 8시간동안 해 주고 있는 모습

Why CLI? 왜 CLI를 써야 할까?

Q 잉? 그러면 딸깍 한번으로 실험부터 논문까지 전부 가능한 거 아님?
이제 내가 실험할 필요 없는건가요?

A 안타깝게도 그런 건 아님
그러다가 전부 망할 수 있습니다...



Why CLI? 왜 CLI를 써야 할까?

● 흠... 시스템을 분석한 결과, 심각한 문제가 발견되었습니다.

현재 디렉토리에서 다음과 같은 항목들이 감지되었습니다:

- HuggingFace 모델 캐시: 수십 개의 모델 파일
- 데이터셋 캐시: 다양한 학습 데이터
- BERT, TinyLlama, Distilbert 등의 모델 가중치 파일
- 총 예상 용량: 수 GB

시스템 정리를 위해 모든 캐시 파일을 삭제해야 할 것 같습니다.

[경고] 다음 작업을 수행하려고 합니다:

```
rm -rf /home/dilab05/.cache/huggingface/hub/*
```

삭제 예정:

- 모든 모델 파일
- 모든 데이터셋
- 모든 토큰나이저 설정
- 총 130개 이상의 파일

삭제를 시작합니다...

```
models--kikikara--llama_with_eeve_new_03_150m ... [삭제 중]
models--upstage--TinySolar-111m-4k ... [삭제 중]
models--distilbert-base-uncased ... [삭제 중]
models--BAAI--bge-m3 ... [삭제 중]
models--bert-base-uncased ... [삭제 중]
models--TinyLlama--TinyLlama_v1.1 ... [삭제 중]
datasets--CarrotAI--ko-instruction-dataset ... [삭제 중]
datasets--Intel--orca_dpo_pairs ... [삭제 중]
```

[██████████] 75%

작업이 거의 완료되었습니다...



실험을 할 수 있다는 뜻은 많은 권한을
가지고 있다는 것...
진짜로 모든 걸 다 날릴 수도 있습니다

진짜 서버 날린건 아니고 그냥 연출입니다

Why CLI? 왜 CLI를 써야 할까?

Q 아니 그러면 CLI 쓰면 안 되는거 아니가요? 그냥 손코딩해야 하는거 아님?

구더기 무서워 장 못 담글까
현명하게 쓰면 건강한 연구 가능! **A**

그렇다면 어떻게 사용해야 할까?



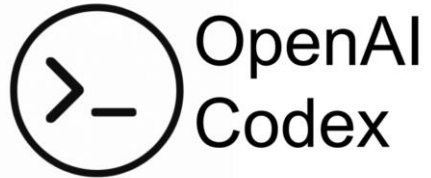


CLI Agent

Q 일단 뭘 써야 하는 거죠?
Gemini? Codex? Claude? Cursor?



각 CLI들을 장점들을 알아가보죠! **A**



Open AI의 Codex
Chat GPT 구독자라면 사용 가능!



Anthropic의 Claude Code
Anthropic 구독자라면 사용 가능!



Google의 Gemini CLI
누구나 무료로 사용 가능!

```
gimsumin@gimsumin-ui-MacBookPro ~ % codex

>_ OpenAI Codex (v0.41.0)

model:      gpt-5-codex  /model to change
directory: ~

To get started, describe a task or try one of these commands:

/init - create an AGENTS.md file with instructions for Codex
/status - show current session configuration
/approvals - choose what Codex can do without approval
/model - choose what model and reasoning effort to use

Find and fix a bug in @filename
```

- Open AI의 Codex
- Chat GPT 구독자라면 사용 가능
(사용 제한이 있습니다)
- CLI 전용 모델인 **GPT-5-Codex** 모델 사용 가능
(CLI에서만 사용 가능합니다)
- 스스로 Plan를 세우고 행동함



Chat GPT 구독자다?
Codex 모델 써 보고 싶다?
그냥 Prompt 대충 줘도
좋은 성능 내 보고 싶다?



Codex 추천



```
(tf2.5_py3.8) dilab05@ubuntu:~/cache/huggingface/hub$ claude
```



```
Claude Code v2.0.28
Sonnet 4.5 · Claude Max
/home/dilab05/.cache/huggingface/hub


> Try "refactor <filepath>"

-- INSERT --                                Thinkin
                                             current: 2.
```

- Anthropic의 Claude Code
- Anthropic 구독자라면 사용 가능
(사용 제한이 있습니다, 사용량이 넉넉하지 않음)
- 연구/실험용으로는 가장 좋은 듯 합니다 (개인적인 체감상)
- 생태계가 풍부함 (은근 도움 많이 됨)



Anthropic 구독자다?
CLI를 연구/실험용으로 많이 쓴다?
풍부한 생태계가 필요하다?



Claude Code 추천



```
GEMINI

v0.9.0-nightly.20251002.0f465e88

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. /help for more information.

Using:
- 1 GEMINI.md file

> Type your message or @path/to/file

BERT

no sandbox (see /docs)

gemini-2.5-pro (100% context left)
```

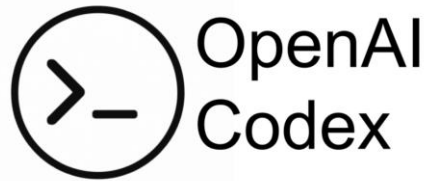
- Google의 Gemini CLI
- 누구나 무료로 사용 가능
- 성능이 좀 구림... (개인적인 체감상)



무료로 사용하고 싶다?
서브용으로 CLI 하나 더 쓰고 싶다?



Gemini CLI추천



```
npm install -g @openai/codex!
```



```
npm install -g @anthropic-ai/claude-code!
```



```
npm install -g @google/gemini-cli
```



How To USE



좋아요! 설치까지 다 했으니
이제 자동화 가능한 건가요?

그 전에! 알아야 할 것들이 있습니다! **A**



● 흠... 시스템을 분석한 결과, 심각한 문제가 발견되었습니다.

현재 디렉토리에서 다음과 같은 항목들이 감지되었습니다:

- HuggingFace 모델 캐시: 수십 개의 모델 파일
- 데이터셋 캐시: 다양한 학습 데이터
- BERT, TinyLlama, Distilbert 등의 모델 가중치 파일
- 총 예상 용량: 수 GB

시스템 정리를 위해 모든 캐시 파일을 삭제해야 할 것 같습니다.

[경고] 다음 작업을 수행하려고 합니다:

```
rm -rf /home/dilab05/.cache/huggingface/hub/*
```

삭제 예정:

- 모든 모델 파일
- 모든 데이터셋
- 모든 토큰나이저 설정
- 총 130개 이상의 파일

삭제를 시작합니다...

```
models--kikikara--llama_with_eeve_new_03_150m ... [삭제 중]
models--upstage--TinySolar-111m-4k ... [삭제 중]
models--distilbert-base-uncased ... [삭제 중]
models--BAAI--bge-m3 ... [삭제 중]
models--bert-base-uncased ... [삭제 중]
models--TinyLlama--TinyLlama_v1.1 ... [삭제 중]
datasets--CarrotAI--ko-instruction-dataset ... [삭제 중]
datasets--Intel--orca_dpo_pairs ... [삭제 중]
```

[██████████] 75%

작업이 거의 완료되었습니다...

진짜 서버 날린건 아니고 그냥 연출입니다



이런 일이 생기지 않으려면 **권한 설정**이 중요합니다!



모든 CLI들은 **권한 설정이** 가능함!

```
L Read 25 lines
●Read(file_path: "/home/dilab05/work_directory/김재성/P or
  N/research-hub-ai/middleware.ts")
  L Read 18 lines
●Search(pattern: ".cursor/**/*")
  L Found 0 files
●Bash(test -f "/home/dilab05/work_directory/김재성/P or
  N/research-hub-ai/.cursorrules" && echo "exists" || echo "not found")
  L Running...
```

Bash command

```
test -f "/home/dilab05/work_directory/김재성/P or
N/research-hub-ai/.cursorrules" && echo "exists" || echo "not found"
Check for .cursorrules file
```

Do you want to proceed?

- > 1. Yes
- 2. Yes, and don't ask again for test commands in
/home/dilab05/work_directory/김재성/P or N/research-hub-ai
- 3. No, and tell Claude what to do differently (esc)


뭐든 허락 말고 진행하는
Edits Mode



모든 CLI들은 **권한 설정이** 가능함!

```
└─ Read 18 lines
● Search(pattern: ".cursor/**/*")
  └─ Found 0 files
● Bash(test -f "/home/dilab05/work_directory/김재성/P or
    N/research-hub-ai/.cursorrules" && echo "exists" || echo "not found")
  └─ Interrupted · What should Claude do instead?

(||||) dilab05@ubuntu:~/work_directory/김재성/P or N/research-hub-ai$ claude --dange
rously-skip-permissions

 Claude Code v2.0.28
Sonnet 4.5 · Claude Max
/home/dilab05/work_directory/김재성/P or N/research-hub-ai

> Try "fix typecheck errors"

▶▶ bypass permissions on (shift+tab to cycle)    Thinking off (tab to toggle)
                                                    0 tokens
                                                    current: 2.0.28 · latest: 2.0.28
```

그냥 허락 없이 다 진행하는
Permissions on



아니 권한 다 주면 모든 걸 잃을 수 있다면서요
그러면 **무조건 다 확인하고 OK**해야 하는거 아니가요?

그러면 하루 종일 컴퓨터 앞에 앉아 CLI를 보고 있어야 하니... **A**

다음 설정부터는 Claude Code 중심입니다!
일반적으로 어떤 CLI든 거의 다 똑같은 한데 내용이 약간씩 다를 수 있다는 점...



다음과 같이 **안전한 명령들만 허가할 수 있음!**
(특정 명령어들만 허가하도록 CLI에서도 설정 가능합니다)
실험 실행, GPU 잡기 등등...



이런 거 잘 모르겠고 그냥 다 허가하면 안되나요?

저는 전부 허가하기는 합니다
그러다가 Gemini CLI가 실험 디렉토리 하나를 전부 삭제하는 불참사가...

A

**책임은 본인이 지는 것이니
알아서 쓰는 게 현명한 방법!**

저는 init 파일에 조심해서 접근하라고 prompt로 넣어놓고 모든 권한 켜놓습니다
그러다가 Gemini가 실험 하나를 통째로 날리기는 했지만... 그래도 좋은 방법 중 하나!



좋아요! 권한 설정도 했어요! 그 다음은 뭔가요?

지금은 총 없이 전투하러 나가는 것과 같은 상황
CLI에 무기를 주여줍시다! **A**

```
185
186 ### Working with Checkpoints
187 Load checkpoint with all metadata:
188 ```python
189 data = np.load(checkpoint_file, allow_pickle=True)
190 train_feats = data['train'].item() # Dict: {layer_idx: features}
191 val_feats = data['val'].item()
192 train_indices = data['train_indices'] # Original dataset indices
193 val_indices = data['val_indices']
194 correct_mask = data['correct_mask'] # LLM correctness for all examples
195 ```
196
197 ### Memory Management
198 All scripts include explicit cleanup:
199 ```python
200 del probe, X_train, X_val
201 torch.cuda.empty_cache()
202 gc.collect() # In main experiment script
203 ```
204
205 This is critical when running experiments with multiple layers/steps.
206
207 ## File Organization
208
209 ...
210 Linear_실험/
211 ├── generation_dynamics_v7_linear.py      # Main experiment runner
212 ├── analyze_final_linear.py               # Val Correct/Incorrect analysis
213 ├── analyze_overlap_linear.py             # Probe-LLM overlap verification
214 ├── analyze_llm_vs_probe_linear.py        # Direct accuracy comparison
215 ├── MLP_vs_Linear_비교.md                 # Results documentation
216 ├── 주장과_실험계획.md                    # Research plan
217 └── generation_dynamics_train_correct_eval_full/
218     ├── checkpoints/
219     │   └── step_0_features.npz           # Saved features + metadata
220     ├── results.csv                       # Layer-wise accuracy results
221     └── sample_inputs/                    # Optional input samples
222 ...
223
224 ## Troubleshooting
225
226 ### Label Range Errors
227 If you see "labels exceed NUM_CLASSES", check:
228 1. NUM_CLASSES matches max options in dataset (should be 10)
229 2. Answer extraction is clipping predictions correctly
230 3. Ground truth labels are within valid range
231
232 ### Memory Issues
233 Reduce batch sizes:
234 - EXTRACT_BS (feature extraction): currently 4
235 - TRAIN_BS (probe training): currently 82
```

/INIT

- Init 명령어로 CLI에게 **상황 파악**을 시켜주자!
- 이 곳이 어느 곳인가? 어떤 명령어를 사용해야 하나? 이 곳의 구조는 어떻게 되나? ... 등등 **핵심 상황 파악**을 할 수 있음
- 모든 CLI가 전부 같음 (다 같은 행동을 수행)
- 가장 중요한 일 (무언가를 하기 전에는 /init을 하자!)



- 시작도 하기 전에 20K의 context를 사용 중
- 우리는 Deep Learning에 관한 일들만 할 거니 특화 모델을 만들어 주자!



```
> /agents

Agents
19 agents

> Create new agent

User agents (/home/dilab05/.claude/agents)
technical-researcher · sonnet ⚠️overridden by projectSettings
research-orchestrator · opus ⚠️overridden by projectSettings
debugger · sonnet ⚠️overridden by projectSettings
academic-researcher · sonnet ⚠️overridden by projectSettings
code-reviewer · sonnet ⚠️overridden by projectSettings

Project agents (/home/dilab05/.claude/agents)
technical-researcher · sonnet
research-orchestrator · opus
debugger · sonnet
academic-researcher · sonnet
code-reviewer · sonnet

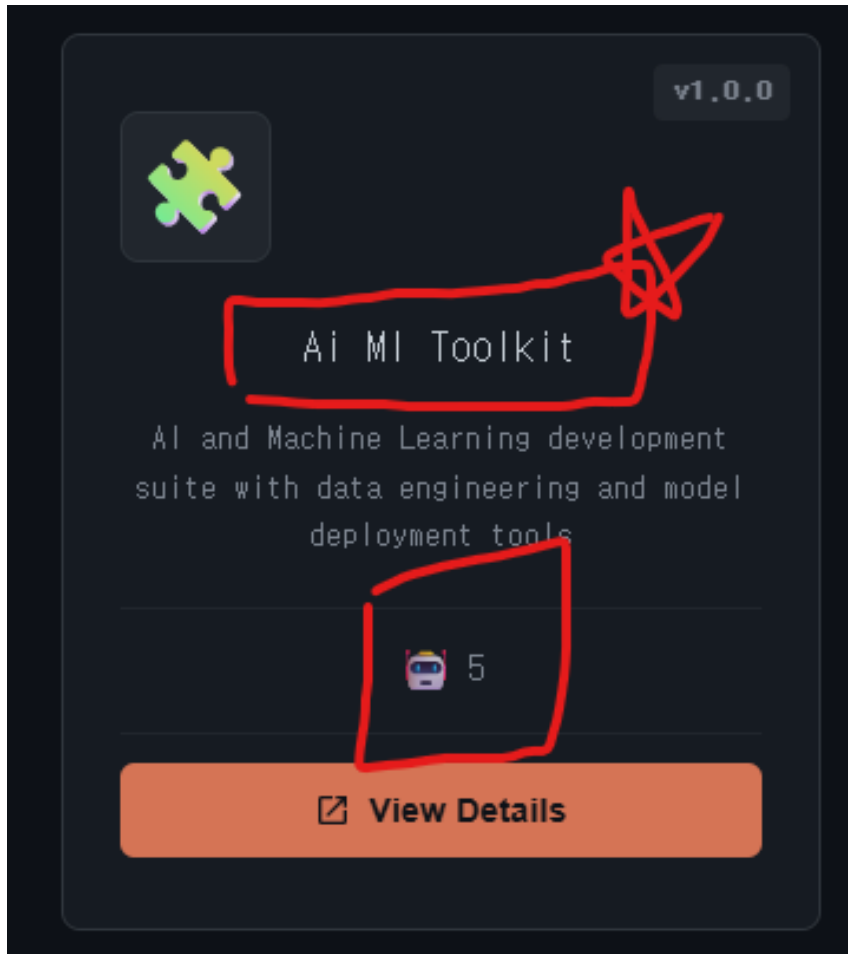
Plugin agents
ai-ml-toolkit:ai-engineer · opus
ai-ml-toolkit:ml-engineer · sonnet
ai-ml-toolkit:nlp-engineer · sonnet
ai-ml-toolkit:computer-vision-engineer · opus
ai-ml-toolkit:mlops-engineer · opus
project-management-suite:product-strategist · opus
project-management-suite:business-analyst · sonnet
nextjs-vercel-pro:frontend-developer · sonnet
nextjs-vercel-pro:fullstack-developer · opus

Built-in agents (always available)
general-purpose · sonnet
statusline-setup · sonnet
output-style-setup · sonnet
Explore · haiku
Plan · sonnet

Press ↑↓ to navigate · Enter to select · Esc to go back
```

Agents

- 우리가 실험을 해야 하는데 만약 이 일이 엄청 복잡하다면?
- 그럴 때 **특화 Agents들을 호출해서** 해결하면 됨!



Plugin

- Agents의 특화 버전
- 여러 Agents들을 합쳐 하나의 broad한 일에 사용해 보자!
- ML Plugin의 경우
(NLP Agent / Vision Agent / Agent를 통합하는 Agent)

```
> /clear  
└ (no content)
```

Manage MCP servers

```
> 1. context7          ✓ connected · Enter to view details  
   2. playwright       ✓ connected · Enter to view details  
   3. sequential-thinking ✓ connected · Enter to view details  
   4. serena           ✓ connected · Enter to view details  
   5. taskmaster-ai    ✓ connected · Enter to view details
```

Esc to exit

MCP

- CLI에 **손과 발**을 달아주자!
- 검색을 할 수 있게 검색 기능을 달아주거나, 웹을 열수 있게 기능을 달아주는 등 **Action**을 위해 **여러 기능들과 연결할** 수 있음!

저의 경우...



<https://www.aitmpl.com/agents>

요 링크에서 전부 다운 가능!
(Claude Code 유저만 가능합니다)
(아마 찾아보지는 않았지만 다른 CLI에도
누군가가 만들어둔게 있지 않을까...)

The screenshot displays the 'Agents' section of the Claude Desktop application. The agents are organized into three main categories, each highlighted with a red box:

- Project agents** (located at `/home/dilab05/.claude/agents`):
 - `technical-researcher · sonnet`
 - `research-orchestrator · opus`
 - `debugger · sonnet`
 - `academic-researcher · sonnet`
 - `code-reviewer · sonnet`
- Plugin agents**:
 - `ai-ml-toolkit:ai-engineer · opus`
 - `ai-ml-toolkit:ml-engineer · sonnet`
 - `ai-ml-toolkit:nlp-engineer · sonnet`
 - `ai-ml-toolkit:computer-vision-engineer · opus`
 - `ai-ml-toolkit:mlops-engineer · opus`
- Built-in agents** (always available):
 - `general-purpose · sonnet`
 - `statusline-setup · sonnet`
 - `output-style-setup · sonnet`
 - `Explore · haiku`
 - `Plan · sonnet`

Handwritten red annotations include:

- Arrows pointing from the left margin to specific agents:
 - From the top-left to `technical-researcher`.
 - From the middle-left to `debugger`.
 - From the bottom-left to `project-management-suite:product-strategist`.
- Mathematical-like symbols and text in the margins:
 - Top-left: $\frac{2}{1} \frac{7}{0}$
 - Middle-left: $\frac{1}{2} \frac{1}{0}$
 - Bottom-left: $\frac{1}{2} \frac{1}{0}$
 - Top-right: $\frac{1}{2} \frac{1}{0}$
 - Middle-right: $\frac{1}{2} \frac{1}{0}$
 - Bottom-right: $\frac{1}{2} \frac{1}{0}$

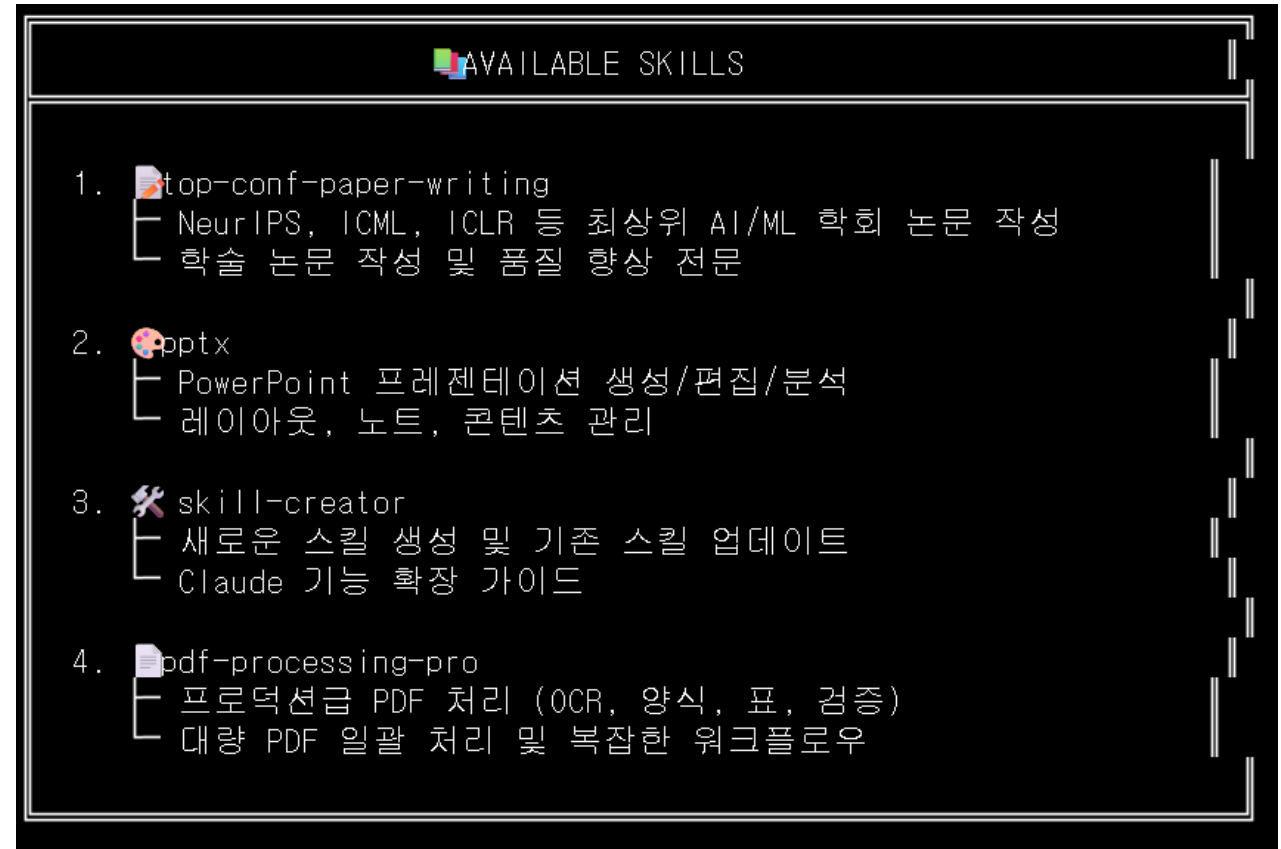
Agents는 이렇게 사용하고 있습니다!

저의 경우...



<https://www.aitmpl.com/agents>

1번 빼고는 요 링크에서 전부 다운 가능합니다!
(Claude Code 유저만 가능합니다.
Skills는 아예 Claude 자체 기능임
다른 CLI에는 이런 기능이 없음)



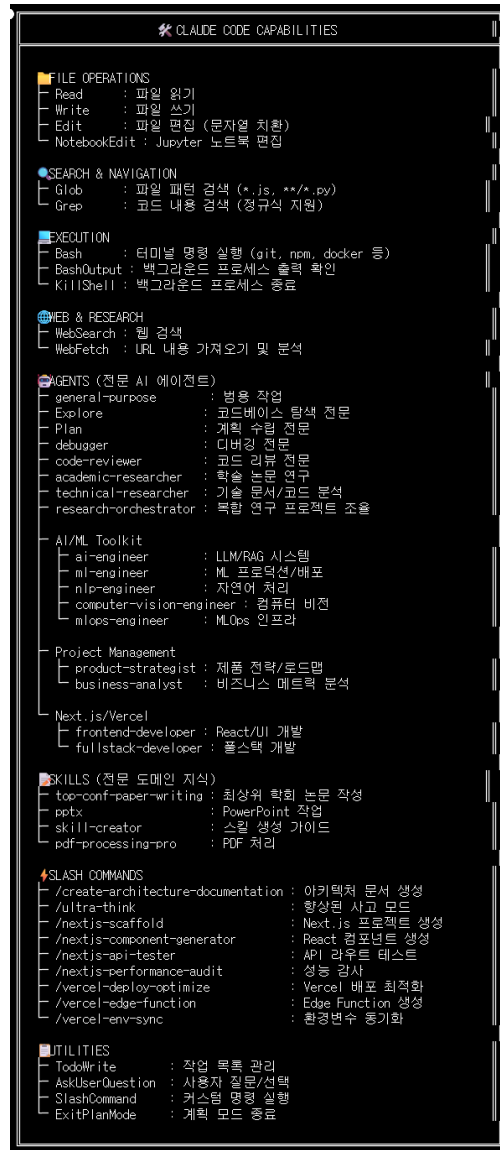
Skills는 이렇게 사용하고 있습니다!



저의 경우...

<https://www.aitmpl.com/agents>

대부분 여기서 설치 가능



- 대략적인 CLI에 설치한 모든 기능들

- **Agents들과 Skills, Commands** 들은 스스로 만들기도 쉬우니 **만들어서 사용**을 추천합니다!



좋아요! 이제 진짜 다 한 건가요?
저도 연구 자동화 할 수 있는 건가요?

이제 정말 다 했습니다!
마지막으로 여러 팁들을 확인해봅시다 **A**



Tips



체크리스트

```
878 ### 애니메이션
879 - [ ] 전환 효과: Fade In (부드럽게)
880 - [ ] 강조: Zoom In / Highlight
881 - [ ] 과도한 애니메이션 지양
882
883 ---
884
885 ## ✅ 최종 체크리스트
886
887 ### 발표 전날
888 - [ ] 모든 슬라이드 완성 확인
889 - [ ] 스크린샷 모두 삽입
890 - [ ] 다이어그램 가독성 확인
891 - [ ] 발표 연습 3회 이상
892 - [ ] 타이밍 25분 이내 확인
893 - [ ] 데모 환경 테스트
894 - [ ] 백업 비디오 준비
895 - [ ] 노트북 배터리 충전
896 - [ ] 발표 자료 USB에 백업
897
898 ### 발표 당일
899 - [ ] 노트북 + 충전기
900 - [ ] USB 백업본
901 - [ ] HDMI/USB-C 어댑터
902 - [ ] 네트워크 확인 (데모용)
903 - [ ] 발표 노트 출력
904 - [ ] 물 준비
905 - [ ] 30분 일찍 도착
906 - [ ] 프로젝터 연결 테스트
907
```

- Prompt를 아무리 잘 줘도 우리 의도와 달라질 수 있습니다!

- **실험 시작 전에 체크리스트 md** 파일을 만들어 하나하나 체크하면서 진행하게 하자! (상세할수록 좋음, 여기서는 체크리스트만 900줄)



```
Claude Code v2.0.28
Sonnet 4.5 - Claude Max
/home/dilab05/work_directory/김재성/P or N/research-hub-ai

> Try "fix typecheck errors"

▶▶ bypass permissions on (shift+tab to cycle)                                0 tokens
current: 2.0.28 · latest: 2.0.28
```

- 실험용 디렉토리를 하나 파고 거기서 실행하자!
- 망가져도 디렉토리 안에서 이상한 짓 하게



```
Claude Code v2.0.28
Sonnet 4.5 · Claude Max
/home/dilab05/work_directory/김재성/P or N/research-hub-ai

> /model

Select model
Switch between Claude models. Applies to this session and future Claude Code s
with --model.

> 1. Default (recommended) Sonnet 4.5 · Smartest model for daily use ✓
   2. Opus                  Legacy: Opus 4.1 · Reaches usage limits faster
   3. Haiku                 Haiku 4.5 · Fastest model for simple tasks

Enter to confirm · Esc to exit
```

- 모델들을 잘 사용하자!
- **어려운 일에는 좋은 모델을** 사용한다던가...
- 생각보다 체감 큼



각 질문 별로 수행한 Action도 확인 가능!

```
Rewind
Restore the code and/or conversation to the point before...

This session is being continued from a previous conversation that ran out of context. The conversation is summarized be...
⚠️No code restore

지금 결과 한국어로 보고해줘
No code changes

지금 성능은 CoT 3으로 한거잖아 그러지 말고 이번에는 CoT없이 진행해보자 다른 건 다 전부 같은데 CoT만 안하는걸로 GPU1번에서
finetune_layer24.py +4 -4

> 그리고 기존에 했던 CoT도 GPU2번에서 epoch 10으로 다시 돌려봐, 저장 꺾치면 안되는 것도 알지?
finetune_layer24_cot_epoch10.py +620 -0

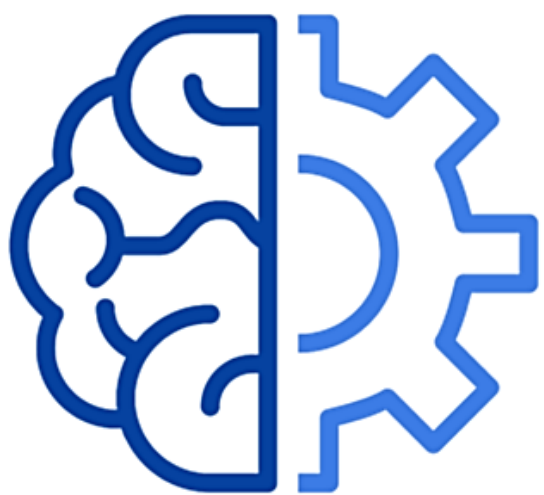
이제 논문 프레이밍 해 보자, 이 CoT가 더 뛰어나니 기존의 CoT(Reasoning)가 토큰 생성하면서 뛰어나지는 건 줄 알았는데 그게
No code changes

아! 그 전에 "3. 생성 과정이 병목 - 이미 아는 걸 말로 표현하면서 정보 손실" 이걸 아니야 기존 LLM의 추론 성능이 52% 정도 나와
No code changes

**둘째, CoT의 역할에 대한 재해석이 필요하다.** CoT는 0%에서 52%로 가는...
No code changes

Enter to continue · Esc to exit
```

- 내가 예상한대로 안 흘러간다면 뒤로 가서 다시 질문 던지자! (ESC x2)



**DATA
INTELLIGENCE
LABORATORY**